UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

| APPLICATION NO. | FILING DATE | FIRST NAMED INVENTOR | ATTORNEY DOCKET NO. | CONFIRMATION NO. |
|---|---|---|---|---|
| 10/692,524 | 10/24/2003 | Steve Johnson | MWS-039RCE | 9823 |

74321 7590 03/30/2009

LAHIVE & COCKFIELD, LLP/THE MATHWORKS
FLOOR 30, SUITE 3000
One Post Office Square
Boston, MA 02109-2127

| EXAMINER | |
|---|---|
| KANG, INSUN | |
| ART UNIT | PAPER NUMBER |
| 2193 | |

| MAIL DATE | DELIVERY MODE |
|---|---|
| 03/30/2009 | PAPER |

**Please find below and/or attached an Office communication concerning this application or proceeding.**

The time period for reply, if any, is set in the attached communication.

| Office Action Summary | Application No. | Applicant(s) | |
|---|---|---|---|
| | 10/692,524 | JOHNSON, STEVE | |
| | Examiner | Art Unit | |
| | INSUN KANG | 2193 | |

*-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --*

**Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

**Status**

1)☒ Responsive to communication(s) filed on *23 December 2008*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

**Disposition of Claims**

4)☒ Claim(s) *1,3,5-16,18-30 and 37-40* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1, 3, 5-16, 18-30, and 37-40* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

**Application Papers**

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

**Priority under 35 U.S.C. § 119**

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

**Attachment(s)**

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

## DETAILED ACTION

1. This action is in response to the amendment filed on 12/23/2008.

2. Claims 1, 3, 5-16, 18-30, and 37-40 are pending in the application.

### *Claim Rejections - 35 USC § 103*

3. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

4. Claims 1, 3, 5-16, 18-30, and 37-40 are rejected under 35 U.S.C. 103(a) as being unpatentable over Conway ("Parsing with C++ Deferred Expressions," ACM SIGPLAN Notices, vol. 29, no. 9, pp. 9-16, ACM, 1994) in view of Walter et al. ("boost C++ Libraries uBLAS Overview," copyright 200-2002) hereafter Walter.

Per claim 1:

Conway discloses:

- providing via a programming language, a language processor with built-in support for a parse tree data structure written in a base language (i.e. page 9, section 2 The deferred expression idiom, second paragraph) said parse tree represented as a class, said class being the basis for a plurality of parse tree objects, said parse tree objects including methods able to retrieve values for base language objects (i.e. page 12, section 5 Embedding Deferred Assignments, third paragraph)
- defining an assignment function, said assignment function taking a plurality of parse

tree structures as arguments (i.e. see page 12, Fig. 4 presenting the deferred assignment operator under section 5. Embedding deferred assignments).

- - defining said assignment function in more than one class (i.e. see page 12, Fig. 4 presenting the deferred assignment operator under section 5. Embedding deferred assignments).

Conway does not explicitly disclose overloading said assignment function based on a context of said base language objects. However, Walter teaches it was known in the pertinent art, at the time applicant's invention was made, to evaluate the left hand side of the overloaded assignment operator. It would have been obvious for one having ordinary skill in the art to modify Conway's disclosed system to incorporate the teachings of Walter. The modification would be obvious because one having ordinary skill in the art would be motivated to make the left-hand size of an assignment available for evaluation based on a context of the objects (i.e. page 4, lines 1-7).

Conway further discloses calling said assignment function to determine the value of at least one assignment within at least one of said base language and a base language extension to said base language (i.e. page 9, section 2 The deferred expression idiom, fourth paragraph).

Per claim 3:

Conway further discloses:

- wherein said assignment function overloads a mathematical operator (i.e. page 13, section 6 Embedding other binary operations, first paragraph).

Per claim 5:

Conway further discloses:

- evaluating said class at compile-time, and (i.e. page 9, section 2. Deferred Expression Idiom, third and fourth paragraphs)
- adjusting the resulting class definitions from said evaluation to increase the efficiency of run-time performance (i.e. page 10, lines 3-6).

Per claim 6:

Conway further discloses:

- overloading a mathematical operator with said assignment function to alter the sequence of evaluation of operands usually followed in said programming language, said overloading designating the order of operand evaluation (i.e. page 12, section 5 Embedding Deferred Assignments, last paragraph).

Per claim 7:

Conway further discloses:

- calling a method in said parse tree object to determine a type of operator at a root of a said parse tree data structure (i.e. page 10, lines 3-6; page 11, second paragraph).

Per claim 8:

Conway further discloses:

- calling a method in said parse tree object to retrieve one of an associated left and right tree (i.e. page 12, section 5 Embedding Deferred Assignments, third paragraph).

Per claim 9:

Conway further discloses:

- wherein the root of said parse tree data structure is one of a constant, variable, a mathematical symbol and a mathematical expression (i.e. page 12, section 5 Embedding Deferred Assignments, second paragraph).

Per claim 10:

Conway further discloses:

- wherein said assignment function is not explicitly defined (i.e. page 12, fig. 4).

Per claim 11:

Conway further discloses:

- wherein said assignment function is used to identify in-place operations (i.e. page 13, fig. 6).

Per claim 12:

Conway teaches the various assignment functions (unary, binary) (i.e. page 13, Fig. 6). Conway does not explicitly teach that said assignment function is used to identify and perform multiply and accumulate ("MAC") operations. However, it would have been obvious for one having ordinary skill in the art to modify Conway's disclosed system to include an assignment

function on MAC operations (A=B+C*D) besides the disclosed operations. The modification would be obvious because one having ordinary skill in the art would be motivated to perform an assignment operation having an MAC operation if present.

Per claim 13:

Conway further discloses:

- wherein said base language is one of C++, Java, System-C, VHDL, Verilog, C#, IDL, MATLAB and a language based on the .Net framework (i.e. page 9, abstract).

Per claim 14, it is another method version of claim 29, respectively, and is rejected for the same reasons set forth in connection with the rejection of claim 29 above.

Per claim 15, it is another method version of claim 30, respectively, and is rejected for the same reasons set forth in connection with the rejection of claim 30 above.

Per claims 16, 18-28, they are other method versions of claims 1-13, respectively, and are rejected for the same reasons set forth in connection with the rejection of claims 1-13 above.

Per claims 29 and 30:

Conway teaches using the parse tree classes to generate code (i.e. page 10, section 3. The ParserExpr Base Class, first paragraph). Conway does not explicitly teach the parse tree classes are used for an embedded processor and processor emulation. However, it would have been obvious for one having ordinary skill in the art to modify Conway's disclosed system to use the parse tree classes in various processor environments if desired. The modification would be

obvious because one having ordinary skill in the art would be motivated to apply the disclosed parse tree structure for an embedded processor and processor emulation, if desired.

Per claims 37 and 38, they are the medium versions of claims 14 and 15, respectively, and are rejected for the same reasons set forth in connection with the rejection of claims 14 and 15 above.

Per claim 39, it is another method version of claim 3, respectively, and is rejected for the same reasons set forth in connection with the rejection of claim 3 above.

Per claim 40:

Conway discloses:

- building a parse tree data structure based on source code with the language processor; (i.e. page 9, section 2 The deferred expression idiom, second paragraph) instantiating a first parse tree object and a second parse tree object (i.e. page 12, section 5 Embedding Deferred Assignments, third paragraph)

- evaluating said second parse tree object to obtain a value (i.e. see page 12, Fig. 4 presenting the deferred assignment operator under section 5. Embedding deferred assignments).

Conway does not explicitly disclose said evaluating done based on a context provided by said first parse tree objects. However, Walter teaches it was known in the pertinent art, at the time applicant's invention was made, to evaluate the left hand side of the overloaded assignment operator. It would have been obvious for one having ordinary skill in the art to modify Conway's disclosed system to incorporate the teachings of Walter. The modification would be obvious because one having ordinary skill in the art would be motivated to make the left-hand

size of an assignment available for evaluation based on a context of the objects (i.e. page 4, lines 1-7).

Conway further discloses assigning said value to said first parse tree object (i.e. page 9, section 2 The deferred expression idiom, fourth paragraph).

***Response to Arguments***

5. Applicant's arguments filed on 12/23/2008 have been fully considered but they are not persuasive.

The applicant states that Conway and Walter alone or in any reasonable combination fail to disclose overloading said assignment function based on a context of said base language objects. Claim 1 does not recite evaluating the left hand side of the overloaded assignment operator as alleged by the examiner. But rather, claim 1 recites overloading said assignment function based on a context of said base language objects. The method of claim 1 evaluates a context such as the size of the variables involved in order to allow context-based assignment function overloading. Nothing in the Walter reference indicates that the context of the evaluated objects should be considered.

In response, the instant specification describes that the context in which the operation is occurring is based on information from the left side of the assignment operator (spec, page 2). The use of the parse tree allows the context to be revealed and the overloading of the assignment function enables the context to become visible (page 3). In Assign (t1, t2), the function "evaluates the value of t2 and assigns the value to the tree object represented by t1 (page 6)." The assignment function, Assign() is defined using two parse tree objects, t1 and t2 as arguments

and the left side of the operator, t1 is obtained by storing the value of left side on a temp variable in the Assign function (page 10). Conway clearly teaches ParserAssignment function corresponding to the instant "Assign" function with two arguments of base language parse objects, lhs and rhs that correspond to t1 and t2 (see Figure 5) storing pointers to the ParserVar for the left-hand side of the operator and the right-hand side (ParserExpr) member (page 12) so that the "deferred assignments may be chained or otherwise embedded in larger expressions (page 12)" from the evaluation of the assignment returning the assigned value. The Evaluate () method performs appropriate context evaluation for returning the left-hand side value in the parser tree in different deferred assignment operations by being redefined for new classes (page 12). Therefore, Conway also discloses a compiler (language processor) that produces parse tree classes (page 9, first par) and the parser assignment function of Conway passes two parse tree objects as arguments. Walter specifically teaches assigning the left hand side of the assignment operator from the evaluation of the right hand side of the assignment into a temporary and then assigning this temp value to the left hand side (i.e. page 4, lines 1-7) as in the instant specification. Therefore, as Conway discloses the corresponding assign function passing two parser tree objects as arguments and Walter specifically discloses evaluating the value of right hand side to assign it to the expression tree object of the left side by using temp, the combination of Conway and Walter disclose overloading said assignment function based on a context of said base language objects.

In response to the application's statement per claims 14-16, 18-30, 37-40 (remark, 11-17), this has been addressed above.

***Conclusion***

6. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

7. Any inquiry concerning this communication or earlier communications from the examiner should be directed to INSUN KANG whose telephone number is (571)272-3724. The examiner can normally be reached on M-R 7:30-6 PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Lewis A. Bullock, Jr. can be reached on 571-272-3759. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300. Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the

Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Insun Kang/
Examiner, Art Unit 2193